```
=> s second (3a) protocol# (3a) stack#
        2019951 SECOND
          95529 PROTOCOL#
         125944 STACK#
L2           61 SECOND (3A) PROTOCOL# (3A) STACK#

=> s first (3a) protocol# (3a) stack#
        2194389 FIRST
          95529 PROTOCOL#
         125944 STACK#
L3           60 FIRST (3A) PROTOCOL# (3A) STACK#

=> s l1 and l2
L4            0 L1 AND L2

=> del l4
DELETE L4? (Y)/N:y

=> s l2 and l3
L4           37 L2 AND L3

=> d 1-37 pn,ab,kwic

L4   ANSWER 1 OF 37  USPATFULL
PI     US 2001047486      A1   20011129
AB     A secure commerce server system and method. A secure commerce server
       system includes a plurality of regions or burbs, including an internal
       burb and an external burb, a commerce server and an administration
       server. Processes and data objects associated with the administration
       server are bound to the internal burb. Processes and data objects
       associated with the commerce server are bound to the external burb.
       Processes bound to one burb cannot communicate directly to processes and
       data objects bound to other burbs. The administration server cannot be
       manipulated by a process bound to the external burb.
CLM    What is claimed is:
 .  .  .  burb cannot communicate directly to processes and data objects bound
       to other burbs, and wherein the internal burb includes a first
       protocol stack and the external burb includes a
       second protocol stack separate from the
       first protocol stack;  a commerce server,
       wherein processes and data objects associated with the commerce server
       are bound to the external burb;  an.  .  .  that a message from the
       commerce server to the administration server is routed up one or more
       layers of the second protocol stack and
       through the assured pipeline to the first protocol
       stack before being routed to the administration server.

 .  .  .  of burbs, wherein the plurality of burbs includes a first and a
       second burb, wherein the first burb includes a first
       protocol stack and the second burb includes
       a second protocol stack separate from the
       first protocol stack;  assigning the
       external network interface to the first burb;  assigning the internal
       network interface to the second burb;  binding processes to burbs;
       establishing an assured pipeline between the first and
       second protocol stacks;  receiving an
       electronic commerce request from the external network;  routing the
       electronic commerce request up one or more layers of the first
       protocol stack and through the assured pipeline to the
       second protocol stack; and  routing the
       electronic commerce request through the second
       protocol stack to the administration server.
```

10. The method according to claim 8, wherein each burb has its own routing table and wherein routing the electronic commerce request up one or more layers of the **first protocol stack** and through the assured pipeline to the **second protocol stack** includes examining an incoming packet to determine if its destination is an address in the first burb's routing table.

L4 ANSWER 9 OF 37 USPATFULL

PI US 6219707 B1 20010417

AB A system and method of achieving network separation within a computing system having a plurality of network interfaces. A plurality of burbs or regions is defined, wherein the plurality of burbs includes a first and a second burb and wherein each burb includes a protocol stack. Each of the plurality of network interfaces is assigned to one of the plurality of burbs and more than one network interface can be assigned to a particular burb. Processes are bound to specific burbs when they try to access that burb's protocol stack and communication between processes assigned to different burbs is restricted so that a communication between a process bound to one burb must pass through a proxy before being sent to a different burb.

CLM What is claimed is:

. . . burbs, wherein the plurality of burbs includes a first and a second burb and wherein the first burb includes a **first protocol stack** and the **second** burb includes a **second protocol stack** separate from the **first protocol stack**; assigning each of the plurality of network interfaces to one of the plurality of burbs, wherein assigning includes assigning the. . . . interface; and if the message is to be routed through the second network interface, routing the message up through the **first protocol stack** to a process bound to the first burb, transferring the message to a process bound to the second burb and routing the message down through the **second protocol stack** to the **second** network interface.

. . . burbs, wherein the plurality of burbs includes a first and a second burb and wherein the first burb includes a **first protocol stack** and the **second** burb includes a **second protocol stack** separate from the **first protocol stack**; assigning each of the plurality of network interfaces to one of the plurality of burbs, wherein assigning includes assigning the. . . . interface; and if the message is to be routed through the second network interface, routing the message up through the **first protocol stack** to a process bound to the first burb, transferring the message through the assured pipeline to a process bound to the second burb and routing the message down through the **second protocol stack** to the **second** network interface.

L4 ANSWER 17 OF 37 USPATFULL
PI US 6085247 20000704
AB A server operating system supports multiple client-server sessions and enables a user to begin a session and later dynamically reconnect to that session even if the user uses two different client computers. The operating system has a multi-user session manager to enable multiple client-server sessions on the server and a multi-user stack protocol manager to manage one or more protocol stacks used in communicating with the clients. When a user connects to the server via a **first** client, the **stack protocol** manager assigns a **first protocol stack** to this **first** client-server connection and the session manager creates a first session for the user. When the user subsequently reconnects to the server using a second client that is different from the first client, the **stack** manager assigns a **second protocol stack** to a **second** client-server connection and the session begins creating a second session for the user. During this latter process, however, the session manager recognizes that the user is affiliated with the first session. The session manager adapts the first session to conform to the system configuration of the second client. The session manager then reassociates the **second protocol stack** with the reconfigured first session so that the user is returned to his/her original session, even though they logged on from a different client.
AB . . . one or more protocol stacks used in communicating with the clients. When a user connects to the server via a **first** client, the **stack protocol** manager assigns a **first protocol stack** to this **first** client-server connection and the session manager creates a first session for the user. When the user subsequently reconnects to the server using a second client that is different from the first client, the **stack** manager assigns a **second protocol stack** to a **second** client-server connection and the session begins creating a second session for the u

L4 ANSWER 27 OF 37 USPATFULL

PI US 5918018 19990629

AB A system and method of achieving network separation within a computing system having a plurality of network interfaces. A plurality of burbs or regions is defined, wherein the plurality of burbs includes a first and a second burb and wherein each burb includes a protocol stack. Each of the plurality of network interfaces is assigned to one of the plurality of burbs and more than one network interface can be assigned to a particular burb. Processes are bound to specific burbs when they try to access that burb's protocol stack and communication between processes assigned to different burbs is restricted so that a communication between a process bound to one burb must pass through a proxy before being sent to a different burb.

CLM What is claimed is:

1. A method of achieving network separation within a computing system having a plurality of network interfaces, including a first. . . wherein the plurality of burbs includes a first and a second burb and wherein the first burb includes a fit **protocol stack** and the **second** burb includes a **second protocol stack** separate from the **first protocol stack**; assigning each of the plurality of network interfaces to one of the plurality of burbs, wherein more than one network. . . interface; and if the message is to be routed through the second network interface, routing the message up through the **first protocol stack** to a process bound to the first burb, transferring the message to a process bound to the second burb and routine the message down through the **second protocol stack** to the **second** network interface.

. . . stack for handling communication across the network interface, wherein each protocol stack is associated with only one burb, wherein a **first protocol stack** is associated with the first burb, and wherein communication from a process bound to the first burb must pass through the **first protocol stack** to a proxy before being sent to the second burb.